ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 8, August 2015

System Verilog Assertion Based Verification

Aswathy Krishnan, Nisha G R

***Abstract*— This Paper describes the assertion based verification for IPs ,using SystemVerilog HDL .Assertion based verification methodology is mainly used for verifying large and complex designs. A separate assertion files created in SystemVerilog are bind with corresponding testbench to validate the design specification and requirements. Any of the specification has been violated one can see the failure messages. 100% efficiency is ensured by using SVA method and analyses are done with Questa Sim EDA tool.**

***Index Terms*— SVA ,IP ,Manchester encoder ,decoder.**

I. INTRODUCTION

Verification Engineers have a few of intention that should be met for respective design**.** These intention or objectives must be composed of design specification, Verification Plan and timing requirements etc .There are lot of techniques an devices used by verification engineer to check their specifications are correct or not. This paper focused on the assertion based verification using SystemVerilog.

An assertion is statement of fact [1]designed by verification engineer for their design validation and specification requirements. It is also known as Random verification method ,which check against the specification of your DUT. If specification requirements are violated ,one can see the failure messages and failure count. Assertion exactly help to increase the productivity of the design or debug and maintained uniform error reporting structure in the simulation part. It can avoid eye balling or wave form comparison as well as time measurements.

Stuart Sutherland et al proposed a SystemVerilog assertions [2]for engineers and try to explore different assertions and test case for the application of DSP designs. Directions are presented for separating assertions between the verification and design engineers.

Black Box verification can be called traditional verification method with black box examining, one can apply test vectors at the primary stimulus of the block without considering the vectors or transactions inside the block, but one can determines the behavior of the primary throughputs .Assertions allows to do white box observability with black box verification. The assertions not only support to detect bugs but also determine the temporal domain condition and corner cases. They are also determining the domain coverage of the test bench as well as design.

Manuscript received Aug 15, 2015

***Aswathy Krishnan**, ,PG Student ,Department of Electronics and Communication Engineering , Mahatma Gandhi University College, Thodupuzha , Kerala,India*
***Nisha G R**, Engineer/scientist,vssc,Trivandrum,Kerala,*

II .Verification Environment

The whole system verification has been characterized with SystemVerilog Assertions. The Intellectual property of core is verified with SVA. Questa Sim EDA tool is used for the realization. Questa Sim supports Verilog, VHDL ,SystemVerilog modules and defined by IEEE standard 1800-2005.It partially supports the following IEEE standard 1800-2009 for SystemVerilog. Questa SIM uses more than one libraries to manage the generation [3] of data before it can used in simulation. Instead of compiling all DUT or Test bench each time you simulate, Questa SIM uses pre-compiled binary data from its libraries. For example, if you make changes to a SystemVerilog Assertion module, it recompiles only that module.

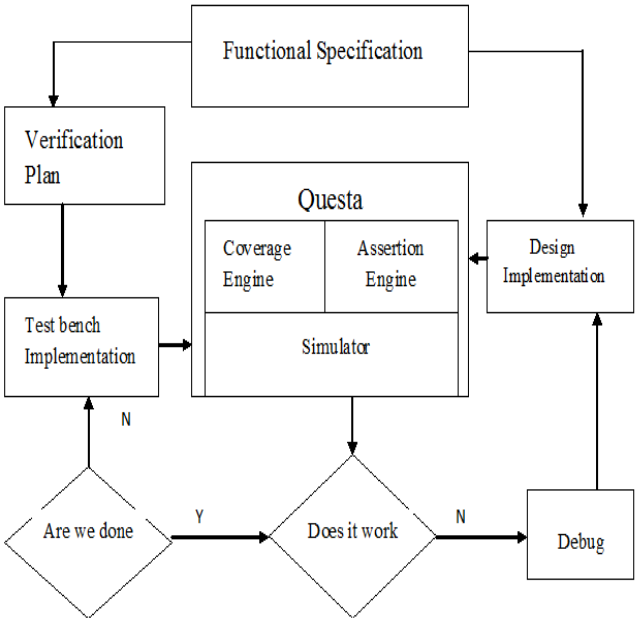

Fig 1 :Verification process

Fig. 1 Shows the Questa Sim verification process . Questa Sim EDA tool supports coverage engine and assertion engine. Coverage engine measures different code coverage and functional coverage. Assertion checks, Where the assertion properties are failed. Binding is possible in Questa Sim. That means assertion are written in the separate file and one can bind the assertion file with signals or ports in the test bench. Multi instance and single instance are the two binding forms.

```
bind encode_1553 :u1 encode_asrt p1(.*);
```
example .1 binding

encode_1553 is the DUT file and encode_asrt is assertion file.

All Rights Reserved © 2015 IJARECE

ISSN: 2278 – 909X
International Journal of Advanced Research in Electronics and Communication Engineering (IJARECE)
Volume 4, Issue 8, August 2015

III. Verification Method

Assertions can be classified into concurrent assertion and immediate assertion. Here each specification has been verified with concurrent assertion property. They may determine the performance or behaviour of the design by using declarations as well as statements. They are specified with implicit operator |=> and disable iff. If the disable iff condition are incorrect, assertion property does not work. Figure (2) shows the SVA verification method. Assumptions are used for the formal verification purpose .Two types of functional verifications are formal verification and hybrid verification. Mainly formal verification methods are used ,because of combinational and temporal domain availability of stimulus.

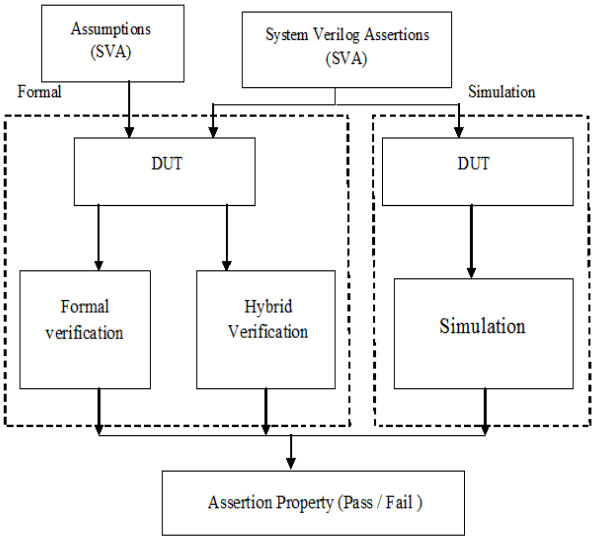

Fig. 2 SVA verification method

Let us consider an example of MIL-STD 15530 Manchester. Design logic is divided into encoder unit and decoder unit [6] .Encoder produce the sync pulse with parity bit as well as encoded data bits. The decoder detects each valid sync and decoding with parity. Here describe the different assertion properties for the verification of Manchester IP core.

```
property ten_ecnt1;
@(posedge c12) disable iff (!reset)
 (ecnt2==ecnt2+1'b1)|=> ##12 ecnt1;
Endproperty
assert property(ten_ecnt1)
else $error("The ten_ecnt1 fails");
```

example .2 Encoder assertion operation

The given example explained that if the reset become true at the posedge of the clock during the evaluation of the sequence then the encoder counter 2 (ecnt2) become ecnt2+1 after the 12 clock cycles of counter1. Otherwise it will never evaluate to true and shows the failure count.Table 1 ,shows the different test plan for the functional verification.

Property name	Description	Remarks
five_eo1	eo1=={(1'b1)} when reset is asserted	Reset condition
six_esm	Assertion of eo1=(1'b1) when(esm=2'b01 && ecnt2=4'h0&& ecnt1=4'h5	Encoder output condition
seven_ee	Assertion of ecnt1=={(4'h0)} when (ee==1)	Enable condition
nine_esm	Assertion of ecnt2==4'h0 when ecnt1=4'h0 && esm=2'b11	Counter condition
ten_ecnt1	Assertion of 12 clock width of ecnt1 when(ecnt2==ecnt2+1'b1)	To ensure 12 clock width
eleven_ee	Assertion pf 5 clock width of serial data out (eo0 & eo1) when ee=0	To ensure 5 clock width of data out

Table 1. Test plan for encoder

Property name	Description	Remarks
three_do1	do16==16'h0000 when reset is asserted	Reset condition
five_dcnt2	dcnt2=={(5'h0)} when reset is asserted	Reset condition
six_dsm	dsm=={(2'b00)} when reset is asserted	Reset condition
seven_do1	Assertion of do16 == dsr when (ds==1'b1&& dcnt2==5'h16 && dcnt3==5'h0)	Output condition
eight_de	Assertion of do16 == dsr when (vw==1'b1)	Valid word condition
nine_de_gne	Assertion of gne when (dcnt2==4'h1 && dcnt1==4'h0)	To ensure 12 clk width of 'gne'

Table 2. Testplan for Decoder

All Rights Reserved © 2015 IJARECE

```
property eight_vw;
@(posedge c16) disable iff (!rst)
( vw == 1'b1) |=> {(dsr == do16)};
endproperty

assert property(eight_vw)
else $error("The eight_vw fails");
```

example 3 Encoder assertion operation

The given example explained that if the reset become true at the posedge of the clock during the evaluation of the sequence .when the valid word is high, decoder 16 bit output(do16) is equal to shift register (dsr). Otherwise it will never evaluate to true and shows the failure count. Table 2 ,shows the different test plan for the functional verification.

IV .Simulation Results

SVA base verification is realized to verify the specification of the Design under Test)DUTs. Functional specification in the design is checked by using assertions. Here we cover two types of simulations results. Any of the specification has been violated, indicates the failure message.

1)Assertion Miss report of the encoder and decoder
2) Assertion Hit report of the encoder and decoder

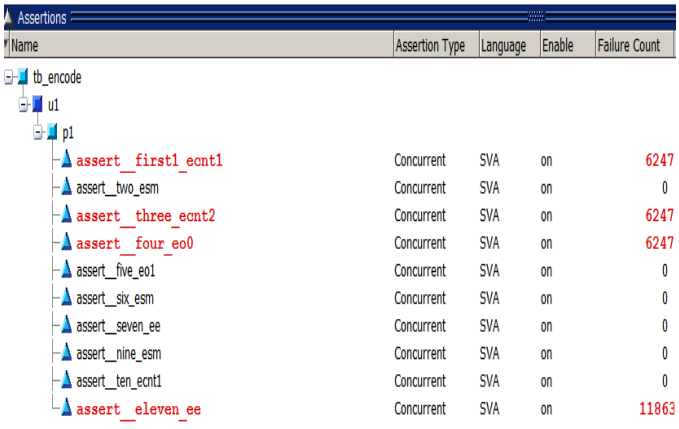

Fig .3(a) Assertions- Miss Report of encoder

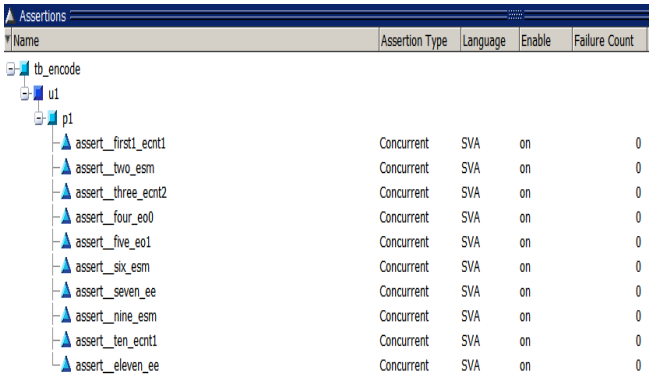

Fig.3(b)Assertions- pass Report of encoder

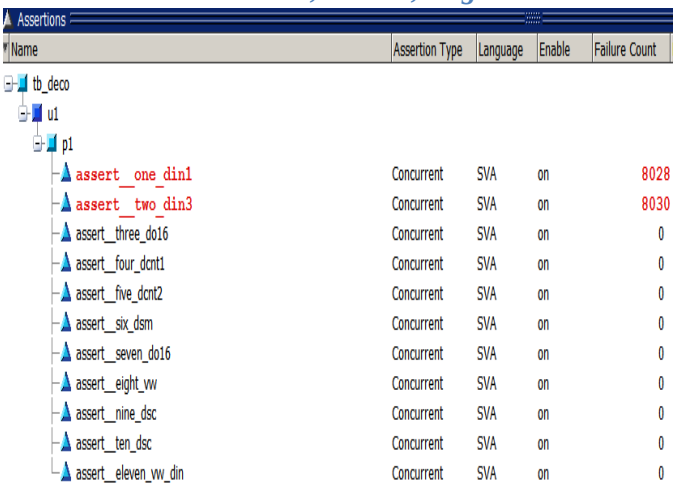

Fig.4(a) Assertions- Miss Report of Decoder

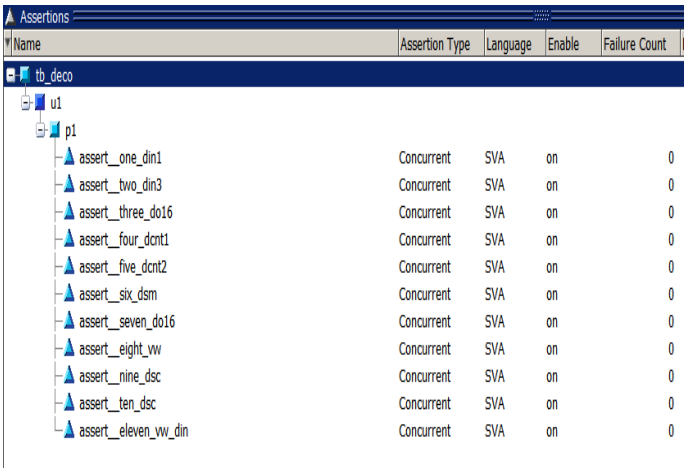

Fig. 4(b)Assertions- pass Report of Decoder

V. Conclusion

In this paper, we discussed the SVA based verification for Manchester Encoder and Decoder using Questa Sim EDA tool. We addressed the topic from the verification engineers viewpoint. From this assertion based verification method 100% efficiency is obtained.SVA is additional indication method to show ,how correctly our entire verification suite exercise the source code. The proposed HDL to verify the IP of 15530 Manchester encoder-decoder logic. This method can be used in any IP verification.

References

[1] Clifford E. Cummings "SystemVerilog Assertions Design Tricks and SVA Bind Files" *SNUG Sanjose* ,2006
[2]Stuart Sutherland, Simon Davidmann .System Verilog For Design, Springer, 2nd Edition ,pp137-166, 2006
[3]Donmills, System Verilog assertion for design engineers , *SNUG Sanjose ,2006*
[4] Questa Sim 10.0 User Manual
[5] Chris Pear , System Verilog For Verification , Springer,2nd Edition,pp.295-302 ,2008
[6]Data sheet on HD-15530

All Rights Reserved © 2015 IJARECE

Aswathy Krishnan , received her graduation in Bachelor of Technology in Applied Electronics and Instrumentation Engineering from Mar Baselios Christian College of Engineering and technology in 2013, Idukki ,kerala. currently PG Scholar at Mahatma Gandhi University College of engineering in Applied Electronics and her research interest include HDL Designs and Control systems.

Nisha G .R ,done the graduation in Bachelor of Engineering in Electronics and Communication Engineering from Bharathiyar University in 1996 ,Tamil Nadu .Joined in VSSC of ISRO in 1997 and acquired 20 years of working experience in the field of design and development of onboard avionics digital systems. Also having the experience in the field of testing and characterization of complex and sophisticated VLSI components like On board FPGA and ASIC designs, Microprocessors ,Microcontrollers ,Memory Devices etc as part of quality assurance programs

All Rights Reserved © 2015 IJARECE